

HTML, CSS & JS basis

Les 1:

Introductie in CSS



Onderwerpen les:

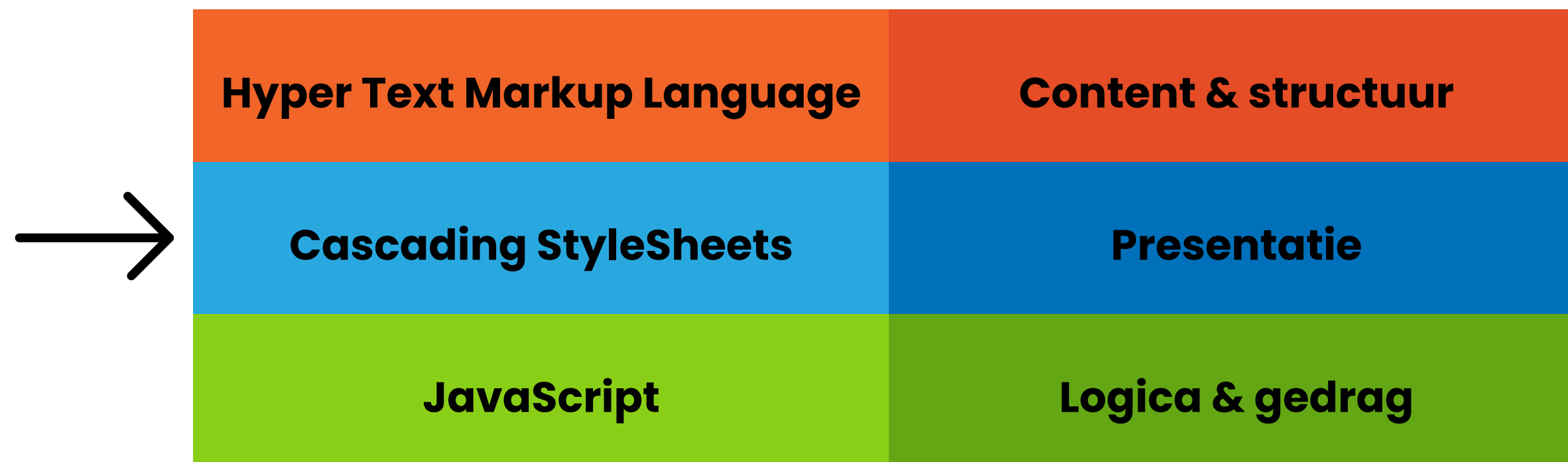
- Wat is CSS?
- Begrippen en Syntax
- Chrome dev tools
- CSS toevoegen aan je pagina
- Selectors (tag, class, id)
- Cascade & Inheritance
- Values & Units (colors, length etc)
- CSS layout (box model, display & position)
- Flexbox

Leerdoelen

- ✓ Aan het einde van de dag ben je bekend met de basis van CSS (Selector, Property & Value).
- ✓ Aan het einde van de dag kan je uitleggen waarvoor we CSS gebruiken.
- ✓ Aan het einde van de dag kan je de meest voorkomende CSS selectors gebruiken.
- ✓ Aan het einde van de dag weet je hoe we kleuren kunnen gebruiken in CSS (hex en rgb).
- ✓ Aan het einde van de dag weet je hoe je relatieve en absolute lengte waarden gebruikt.
- ✓ Aan het einde van de dag weet je het verschil tussen position relative en absolute.
- ✓ Aan het einde van de dag kan je het boxmodel uitleggen.
- ✓ Aan het einde van de dag weet je wat flexbox is.

Front end development

De drie lagen van webdevelopment



CSS: hierin wordt de presentatie van de content bepaald. Welke kleur heeft de tekst? Hoe breed wordt een blok? Heeft de achtergrond een kleur of een plaatje? etc.

Wat is CSS?

Cascading Style Sheet

Trapsgewijs opmaak model. Hiermee selecteer je elementen in je HTML pagina's en verander je hun presentatie.

De presentatielaag van de presentatielaag. Css zorgt voor de styling van de elementen op een pagina. CSS is een makkelijk te leren programmeertaal veel van de codes logische benamingen hebben. Zoals width, height, color, font-size etc. In CSS selecteer je een HTML element en geef je deze stylingsregels.

Begrippen en Syntax

Selector

Selectoren gebruiken we om HTML-elementen te selecteren. (bv: de naam van een tag/class/id)

Property

Properties geven aan wat we willen aanpassen. (bv: color of width)

Value

Value is de waarde die wij toekennen aan een property. (bv: red, 100px)

Begrippen en Syntax

Syntax

```
selector {  
  property: value;  
}
```

Een element selecteer kan op 3 manieren:
aan de hand van een tag, class of id. In het code blok
omringd door curly brackets volgen de property en
value.

Begrippen en Syntax

Syntax voorbeelden:

```
h1 {  
  max-width: 720px;  
}
```

```
#idname {  
  max-width: 720px;  
}
```

```
.classname {  
  max-width: 720px;  
}
```


Chrome dev tools

Chrome dev tools

Voor debuggin en leuk om mee te experimenteren! Controleer HTML CSS en Javascript en haal bugs uit je website. Verander CSS live en bekijk het resultaat direct! Bekijk HTML en CSS van andere websites om te zien hoe iets is gebouwd.

Openen DevTools:

rechter muisklik -> inspecteren



Chrome dev tools

HTML CSS:

Tabblad **Elements** -> hier zie je het Document Object Model met alle HTML elementen.

In het submenu zie je onder het tabblad **Styles** de css styling van het geselecteerde element.

Javascript:

Tabblad **Console** -> hier zie je de console.log van de pagina en eventuele issues en errors.

| https://www.w3schools.com/js/js_htmlDOM.asp

CSS toevoegen

CSS kan op 3 plekken worden geplaatst: inline, internal stylesheets en external stylesheets.

Inline

Hierbij plaats je de css als attribuut van een element in de html zelf.

Internal Stylesheet

Dit is een stylesheet in de head van de pagina zelf. Dit doe je door in de head style tags te plaatsen met daarin de css regels.

External Stylesheet

Dit is een stylesheet in een apart css bestand waarnaar gelinkt wordt vanuit de pagina.

CSS toevoegen

Inline

```
<h1 style="color:blue;">A Blue Heading</h1>
```

Internal Stylesheet

```
<head>  
<style>  
h1 {  
    color: blue;  
}  
</style>  
</head>
```

CSS toevoegen

External Stylesheet

```
<head>  
<link rel="stylesheet" href="mystyle.css">  
</head>
```

In mystyle.css

```
h1 {  
  color: blue;  
}
```

| https://www.w3schools.com/css/css_howto.asp

CSS selectors basis

3 basis manieren

tag

```
h1 {}
```

class

```
.classname {}
```

id

```
#tagname {}
```

CSS selectors basis

Meer manieren

```
* {}
```

Selecteer alle elementen

```
h1, h2, .classname {}
```

Meerdere tags, classes of id's (scheiden met komma)

Selecteer alle h1, h2 en .classname

```
div p {}
```

Selecteer elementen in een ander element

Selecteer alle p's in een div. Hoeft geen direct child te zijn

```
div > p {}
```

Selecteer elementen in een ander element

Selecteer alle p's in een div. Alleen direct childs

| https://www.w3schools.com/csSref/css_selectors.php

Cascade & Inheritance

Vorrang bepalen (cascade)

Meerdere bronnen

Styling komt dus uit meerdere bronnen en via de author styling kan een element ook nog eens op verschillende manieren op plekken op verschillende bestyled worden.

Bron van minst belangrijk naar belangrijkst

User agent stylesheets	Styling van de browser
------------------------	------------------------

User stylesheets	Styling van de user
------------------	---------------------

Author stylesheets	Styling van de author
--------------------	-----------------------

Cascade & Inheritance

Vorrang bepalen (cascade)

Meest specifieke selector

In de stylesheet van de author kunnen op meerdere plekken worden verwezen naar het zelfde element. Bijvoorbeeld via zijn tag, class of id.

Specificiteit van minst belangrijk naar belangrijkste

1 tag (minst belangrijk)

2 class

3 combo class

4 id

5 inline styling

* !important overruled alles

Inheritance

Overerven van styling

Inheritance is het overerven van styling van parent elements. Niet alle styling wordt overgeërfd. Bepaalde properties worden wel overgenomen en anderen niet.

Voorbeeld waarbij het wel gebeurd zijn: color, font-size en line-height.

Wat houdt dit in? Ik heb een div met de class: color-red met daarin een h2 en een p tag. Deze h2 en p geef ik geen styling maar de parent .color-red geef ik color red. Nu overerven deze 2 element ook de kleur rood.

Inheritance

Voorbeeld inheritance

```
<div class="color-red">  
  <h2>Heading tekst</h2>  
  <p>Lorem ipsum dummy tekst</p>  
</div>
```

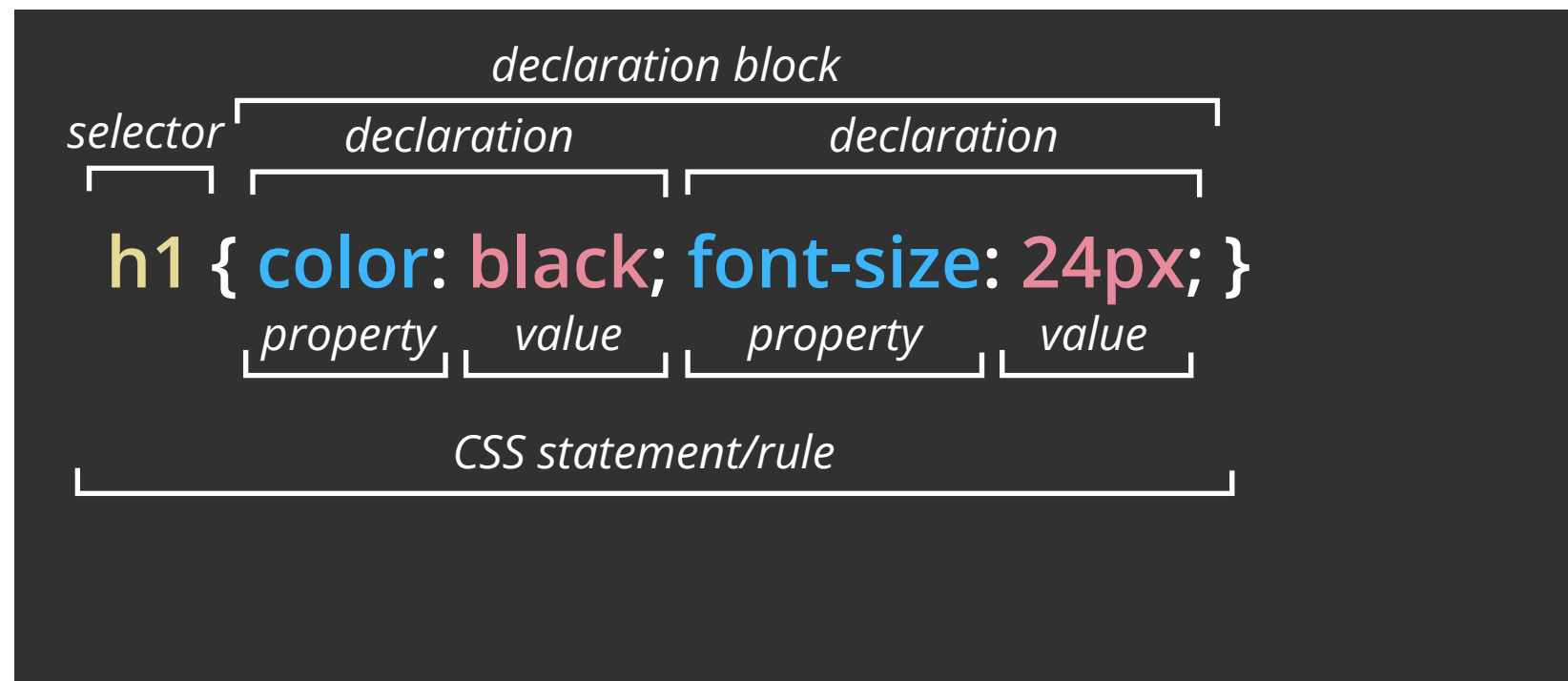
```
.color-red {  
  color: red;  
}
```

| https://developer.mozilla.org/en-US/docs/Learn/CSS/Building_blocks/Cascade_and_inheritance

Values & Units

CSS values

Een CSS value is de waarde die je aan een property toekent binnen het declaration block.
Dit bij elkaar noemen we een CSS rule of CSS statement.



Values & Units

Integeres & numbers

Integers

Het integer data type bestaat uit hele getallen 0 t/m 9. Integers worden gebruikt bij z-index, bij flex en grid instellingen en nog een paar andere css properties.

```
.my_element {  
  z-index: 99;  
}
```

Numbers

Het numbers data type kunnen in tegenstelling tot Integers wel getallen achter de komma bevatten. Ook worden wetenschappelijke notaties geaccepteerd.

```
.my_element {  
  opacity: 0.2;  
  line-height: 1.4;  
}
```

Values & Units

Dimensions & Percentages

Dimensions

Een dimension is een data type bestaande uit een number gecombineerd met een meet unit zoals px of ms. Een dimension kan een lengte, tijd, frequentie of resolutie aanduiden.

```
.my_element {  
  width: 100vw;  
  max-width: 600px;  
  transition: all 20ms  
}
```

Percentages

Percentage geeft een waarde relatief ten opzichte van zijn parent. Als je een width zet op 50% dan wordt het element 50% breed ten opzichte van zijn parent.

```
.my_element {  
  width: 50%;  
}  
  
.my_parent_element {  
  width: 400px;  
}
```

Values & Units

Textuele data

Strings

Een String bestaat uit unicode karakters omringd door dubbele of enkele quotes. Het wordt onder andere gebruikt voor de content, font-family en quotes property.

```
.my_element {  
  font-family: 'Poppins';  
}
```

Urls

De url() wordt gebruikt om een bestand toe te voegen zoals een afbeelding. Deze url kan een absolute of een relatieve url bevatten.

```
.my_element {  
  background-image: url("image.png");  
}
```

Values & Units

Length units: relatief en absoluut

Relatieve units zijn een manier van het meten van een afstand relatief aan een andere afstand.

Absolute units hebben een vaste waarde zoals centimeters inches en pixels.

%

Relatief aan parent element

em

Relatieve aan de font-size van het element

rem

Relatief aan de font-size van het root element

vh

Relatief aan x% van de hoogte van de viewport

vw

Relatief aan x% van de breedte van de viewport

px

Absoluut: Pixels zijn afhankelijk van het viewing device

Values & Units

Color units

Een color unit geeft een kleur aan, deze kleur kan worden aangegeven met een keyword zoals blue, black of zelfs transparent, een rgb waarde als een hex notatie of een rgb()/rgba() functie en als een hsl()/hsla() functie.

Color keywords

Het gebruik van color keywords is erg makkelijk. Zoals je in de lijst op mozilla ziet heb je een erg uitgebreide keuze aan keywords.

```
.my_element {  
  color: blue;  
  background-color: black;  
}  
  
.my_element2 {  
  color: black;  
  background-color: transparent;  
}
```

Values & Units

RGB

Een hexadecimale kleur is gespecificeerd met #RRGGBB (RR (red), GG (green) en BB (blue)). De rgb() maakt gebruik drie getallen waarmee wordt aangegeven de intensiteit van rood, groen en blauw met als waarden – tm 255. rgba() voegt daar alpha aan toe en dat staat voor de transparantie en wordt uitgedrukt in een 0.0 tot 1.0.

```
.my_element {  
  color: red;  
  color: #fc0000;  
  color: rgb(255,0,0);  
  color: rgba(255,0,0,1.0);  
}
```

Rood op vier verschillende manieren

Values & Units

Functions

rgb() en rgba() zijn eigenlijk ook functies voor het berekenen van een kleur. In CSS kunnen heel veel functies worden gebruikt deze zijn op te delen in de volgende categorieën: transform functions, math functions, filter functions, color functions, image functions, counter functions, font functions en shape functions.

Transform functions

Deze functies worden gebruikt om een element te transformeren zoals: roteren, schalen, verplaatsen en schuintrekken.

```
.my_element {  
  transform: scale(1.2);  
}  
  
.my_element2 {  
  transform: rotate(45deg);  
}
```

Values & Units

Functions

Filter functions

Deze functies worden gebruikt om een grafische effect toe te passen op een element zoals: blur, opacity, sepia en meer.

```
.my_element {  
    filter: blur(4px);  
}  
  
.my_element2 {  
    filter: sepia(60%);  
}
```

Values & Units

Functions

Math functions

Met math functions kun je nummers noteren als wiskundige expressies. Je kan onder andere gebruiken `calc()`, `max()`, `min()` en meer.

```
.my_element {  
  width: calc(50% + 100px);  
}  
  
.my_element2 {  
  width: max(20vw, 100px);  
}
```

Values & Units

Functions

Image functions

Onder image functions vallen `image()` maar ook gradient functies zoals `linear-gradient()`. Deze functies kunnen overal gebruikt worden waar een afbeelding een geldige value is.

```
ul {  
  list-style-image: image(ltr 'img.png');  
}  
  
.my_element2 {  
  background: linear-gradient(#e66465, #9198e5);  
}
```

Values & Units

Extra informatie

- | <https://developer.mozilla.org/en-US/docs/Web/CSS/integer>
- | <https://developer.mozilla.org/en-US/docs/Web/CSS/number>
- | <https://developer.mozilla.org/en-US/docs/Web/CSS/dimension>
- | <https://developer.mozilla.org/en-US/docs/Web/CSS/percentage>
- | <https://developer.mozilla.org/en-US/docs/Web/CSS/string>
- | [https://developer.mozilla.org/en-US/docs/Web/CSS/url\(\)](https://developer.mozilla.org/en-US/docs/Web/CSS/url())
- | https://www.w3schools.com/cssref/css_units.asp
- | <https://developer.mozilla.org/en-US/docs/Web/CSS/length>

Values & Units

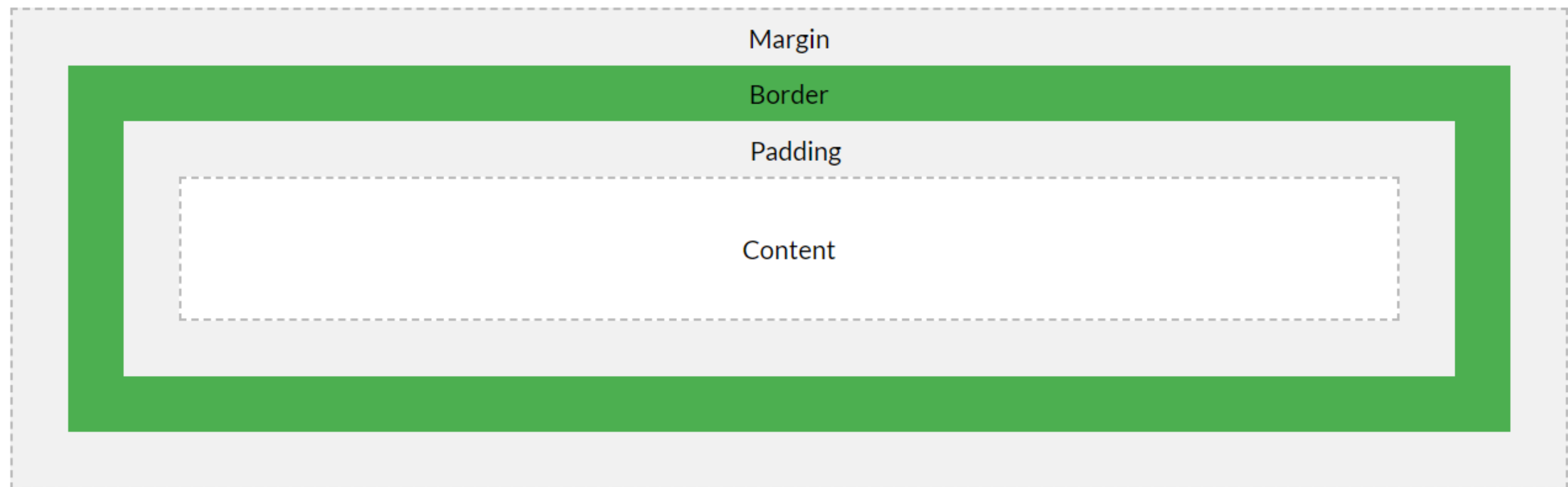
Extra informatie

- | https://developer.mozilla.org/en-US/docs/Web/CSS/color_value
- | https://www.w3schools.com/css/css_colors.asp
- | <https://colors.co/>
- | https://developer.mozilla.org/en-US/docs/Web/CSS/CSS_Functions

CSS layout

Box model

Een website bestaat eigenlijk uit allemaal blokken in blokken. Deze blokken 'boxes' bestaan vervolgens zelf uit content, padding, border en een margin.



CSS layout

Box model

Content	de inhoud van de box, dit kan tekst zijn of een afbeelding, maar ook andere boxes.
Padding	de ruimte in de box om de content heen. Geef je het element een achtergrond kleur? dan verandert het gedeelte van de padding ook.
Border	de rand om de content en de padding. Deze rand kan je een solide kleur geven, maar ook een dotted line. Ook kan je hiermee een element een afgerond erand geven.
Margin	de ruimte tussen dit element (box) en het element ernaast, boven of onder.

| https://www.w3schools.com/css/css_boxmodel.asp

CSS layout

box-sizing: border-box

De width en height van een element worden standaard toegepast alleen op de content van de box. Als het element een padding heeft wordt deze toegevoegd aan de width en de height van een element.

De box-sizing property kan dit aanpassen zodat de padding en border onderdeel wordt van de width en height van het element.

```
.my_element {  
  box-sizing: border-box;  
}
```

| https://www.w3schools.com/css/css3_box-sizing.asp

CSS layout

Padding

Een padding wordt gebruikt om ruimte omde content te creeren. De padding zit binnen de border.

Padding instellen op een element.

De padding instellen kan met de property padding, maar ook met de properties: padding-top, padding-bottom, padding-left, padding-right.

```
.myelement {  
  padding: 20px;  
}
```

1 waarde: boven, onder, links en rechts

```
.myelement {  
  padding: 20px 10px;  
}
```

2 waarden: 1e waarde staat voor boven en onder, 2e staat voor links en rechts

CSS layout

Padding

```
.myelement {  
  padding: 20px 40px 10px 60px;  
}
```

4 waarden: 1e is boven, 2e is rechts, 3e is onder, 4e is links

```
.myelement {  
  padding-top: 20px;  
}
```

De padding wordt toegepast aan de aangegeven kant.
Top, right, bottom of left

| https://www.w3schools.com/css/css_padding.asp

CSS layout

Margin

Een padding wordt gebruikt om ruimte buiten het element te creëren. De margin zit buiten de border.

Margin instellen op een element.

De margin instellen kan met de property margin, maar ook met de properties: margin-top, margin-bottom, margin-left, margin-right.

```
.myelement {  
  margin: 20px;  
}
```

1 waarde: boven, onder, links en rechts

```
.myelement {  
  margin: 20px 10px;  
}
```

2 waarden: 1e waarde staat voor boven en onder, 2e staat voor links en rechts

CSS layout

Margin

```
.myelement {  
  margin: 20px 40px 10px 60px;  
}
```

4 waarden: 1e is boven, 2e is rechts, 3e is onder, 4e is links

```
.myelement {  
  margin: 20px;  
}
```

De padding wordt toegepast aan de aangegeven kant.
Top, right, bottom of left

| https://www.w3schools.com/css/css_margin.asp

CSS layout

Border

Een border is de rand om de content + padding. De border bestaat uit een style, width en een kleur. Daarnaast kan je ook de radius (ronding) instellen.

Border instellen op een element.

De border kan worden ingesteld met de border-style, border-width, border-color en dit kan ook per kant gedaan worden zoals border-top-style etc. Ook is er een shorthand voor style, width en color namelijk: border.

border shorthand

```
.myelement {  
  border: 4px dotted red;  
}
```

border shorthand per kant

```
.myelement {  
  border-left: 4px dotted red;  
}
```


CSS layout

Border

border-style

```
.myelement {  
  border-style: dotted;  
}
```

border-width

```
.myelement {  
  border-width: 4px;  
}
```

border-color

```
.myelement {  
  border-color: red;  
}
```

border-style per kant

```
.myelement {  
  border-left-style: dotted;  
}
```

| https://www.w3schools.com/css/css_border.asp

CSS layout

Display

Met de property display wordt bepaald wat voor display gedrag een element krijgt. (type box dat getoond wordt). Krijgt het element een nieuwe regel? Krijgt een element alleen de ruimte die nodig is of een hele regel?

Display values

Op w3schools zie je een hele lijst met mogelijkheden! In deze les wordt behandeld: inline, inline-block, block, flex en none.

Inline

Wordt getoond als een inline element. Het element krijgt alleen de ruimte die nodig is en wordt niet geplaatst op een nieuwe regel. Height en width properties hebben geen effect.

Inline-block

Is hetzelfde als een inline element echter hier kan je wel height en width properties op toepassen.

CSS layout

Display

Block

Wordt getoond als een block element. Het element krijgt een nieuwe regel en neemt de gehele regel in beslag.

None

Het element wordt niet getoond.

Flex

Zie hoofdstuk flexbox.

| https://www.w3schools.com/cssref/pr_class_display.asp

CSS layout

Position

De property position regelt de positionering van een element. De default waarde is static en worden niet beïnvloed door de properties top, bottom, left en right.

Position values

Static

Een element wordt gepositioneerd volgens de normale flow van de pagina. Properties als top bottom, left en right hebben geen invloed.

Relative

Een element krijgt zijn normale positie net als bij static, maar kan wel worden beïnvloed door top, bottom, left en right properties.

Absolute

Een element wordt gepositioneerd aan de hand van het eerste parent element die niet position static heeft als position waarde. Het element wordt gepositioneerd aan de hand van top, bottom, left en right properties.

CSS layout

Position

Fixed

Het element wordt gepositioneerd ten opzichte van de viewport, dit betekent dat het element altijd op dezelfde plek blijft tijdens het scrollen. De properties top, bottom, left en right bepalen waar het element op de viewport geplaatst worden.

Sticky

Het element switched tussen relative en fixed, afhankelijk van de positie van de viewport.

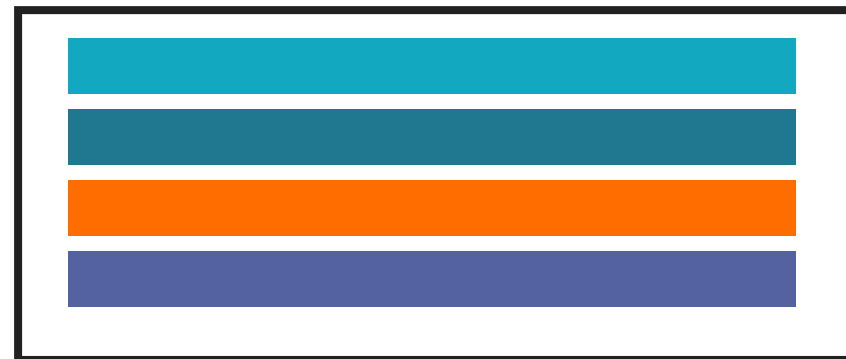
| https://www.w3schools.com/css/css_positioning.asp

CSS layout

Flexbox

Flexbox is een geweldige layout display setting, waar je makkelijk responsive websites mee maakt.

Met flexbox kun je de richting, grootte en volgorde bepalen van elementen. Hiermee zet je makkelijk elementen naast elkaar, creëer je kolommen en maak je je website responsive.



standaard display block



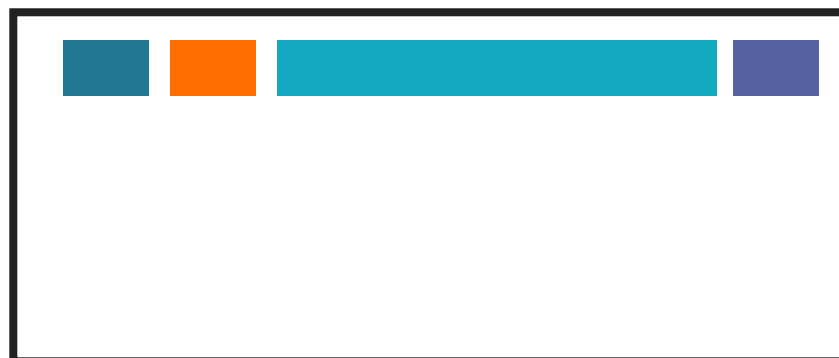
display flex



display flex variant



display flex variant



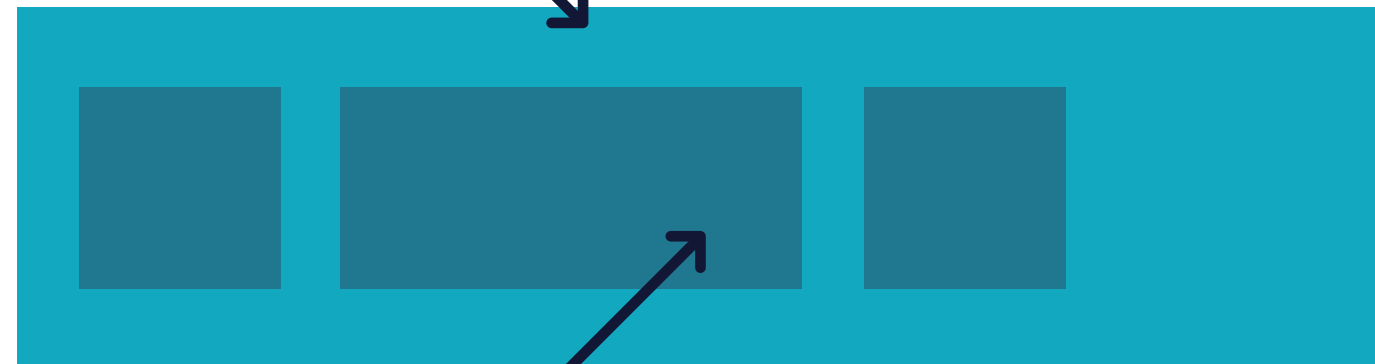
display flex variant

CSS layout

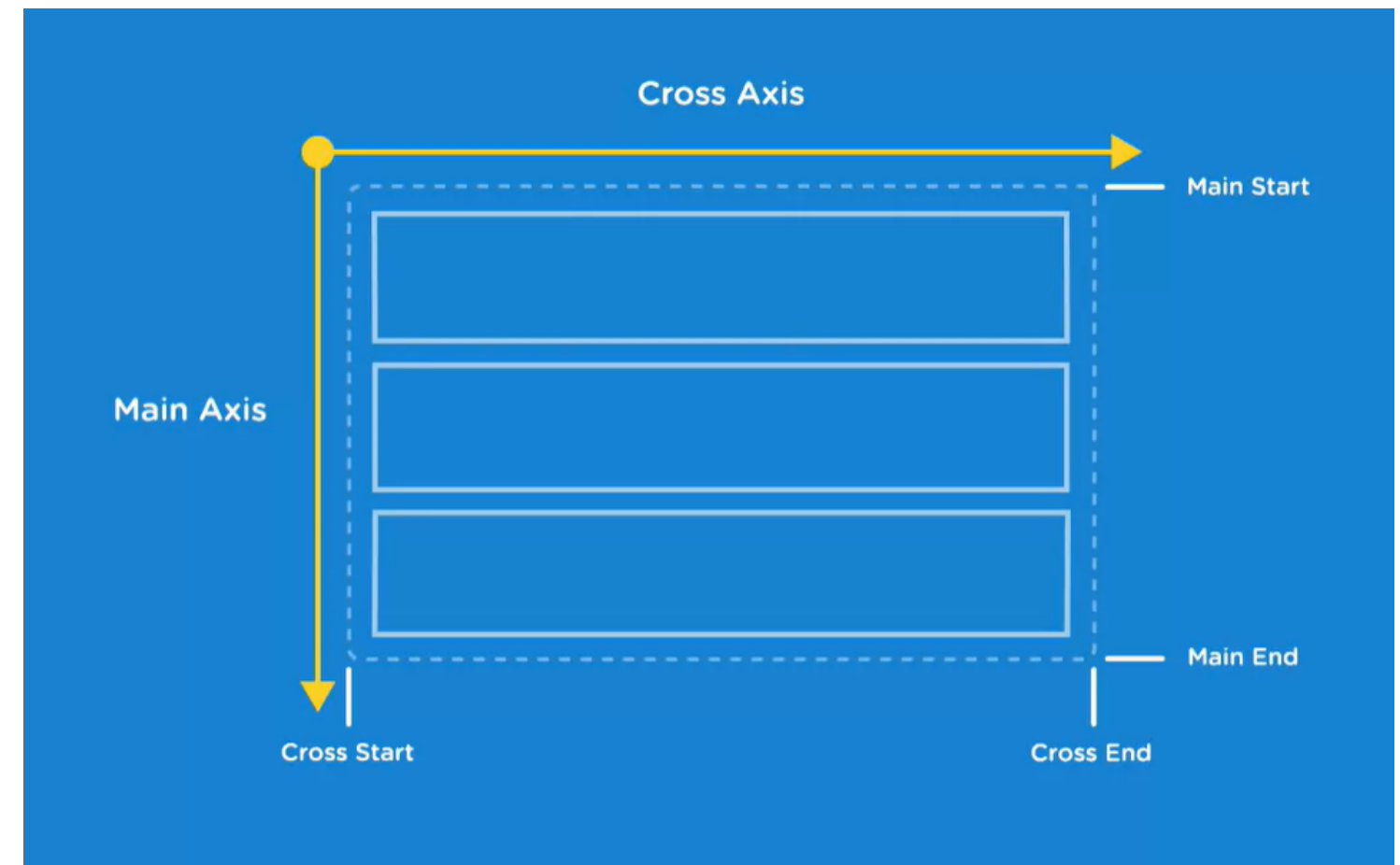
Flexbox

Het element met display flex wordt de flexcontainer
De direct child elementen in de flexcontainer worden de flex items genoemd.

flex container



flex items



CSS layout

Flexcontainer

- Element stylen met display: flex.
- Is een wrapper waar items in zitten.

flex-direction

Met flex-direction stel je de richting in van de elementen in de flex container. Wij gaan gebruiken row of column. row is de default waarde (alles komt naast elkaar), bij column komt alles onder elkaar:

```
.naam-class {  
  display: flex;  
  flex-direction: row;  
}
```

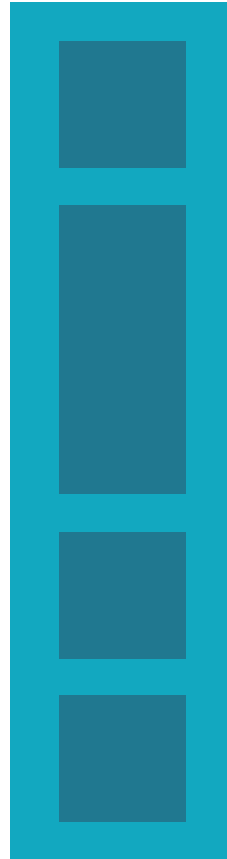

CSS layout

flex-direction

row



column



Values:

row
row-reverse
column
columnreverse

CSS layout

flex-wrap

flex items proberen op 1 regel te passen. Dit kan resulteren in te kleine items en is het beter dat de items springen naar een nieuwe regel zodra de regel vol is. In dit geval is flex-wrap de oplossing.

wrap



Opties:

- nowrap (default)
- wrap
- wrap-reverse

CSS layout

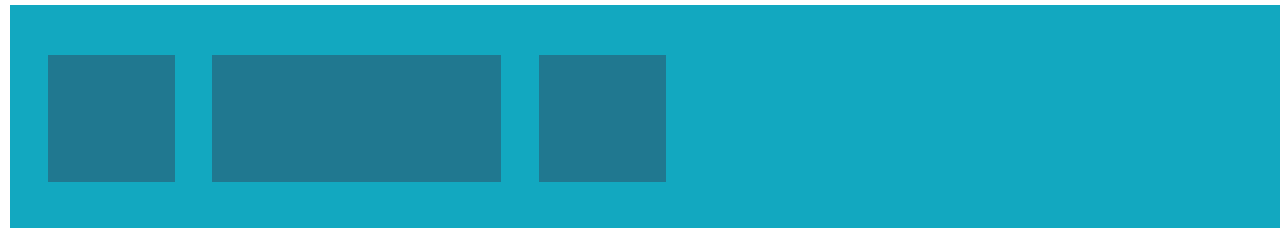
justify-content

De alignment langs de main axis.

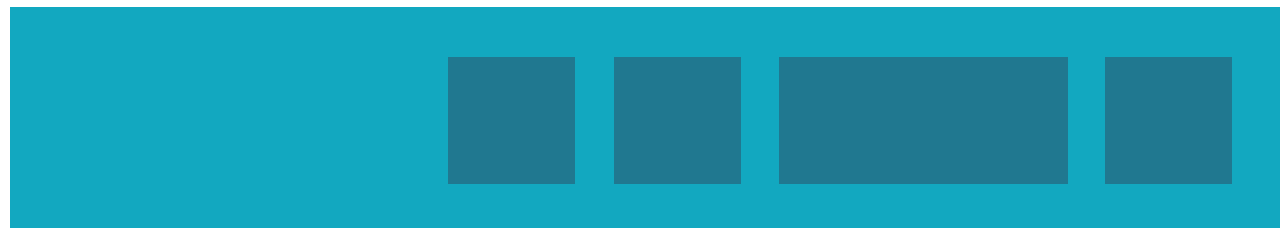
Opties:

flex-start (default)
flex-end
flex-center
space-between
space-around
space-evenly

flex-start



flex-end



flex-center



space-around



space-evenly



space-between



CSS layout

align-items De alignment langs de cross axis.

Opties:
flex-start
flex-end
center
stretch (default)
baseline

flex-start



flex-end



flex-center



stretch



CSS layout

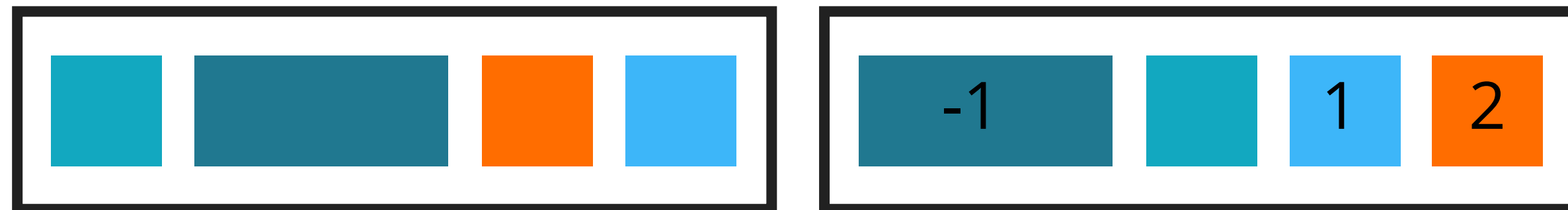
Flexitems

- Elke direct child in een flex container is een flex item.
- Je kan zoveel flex items in een flex container plaatsen als nodig is.

order

Default = 0: Items met dezelfde waarde worden geplaatst in de volgorde waarin ze in de HTML geplaatst zijn.

De volgorde kan worden aangepast met positieve of negatieve waarden (-1, 1 of 3 etc). De laagste waarden worden als vooraan geplaatst.

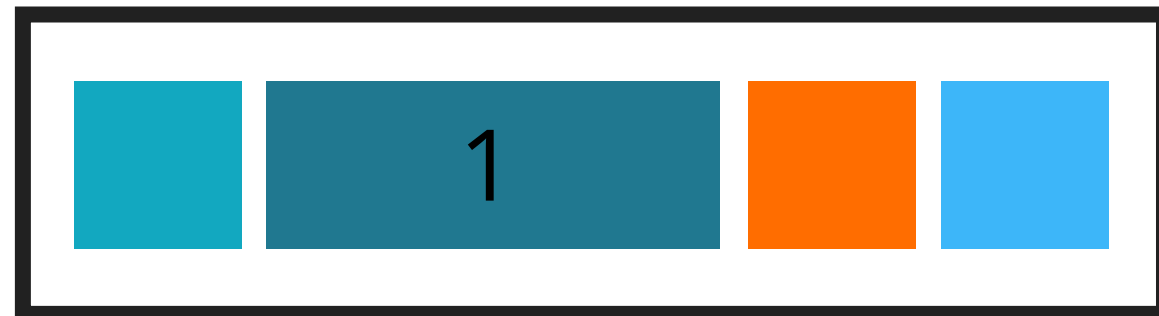
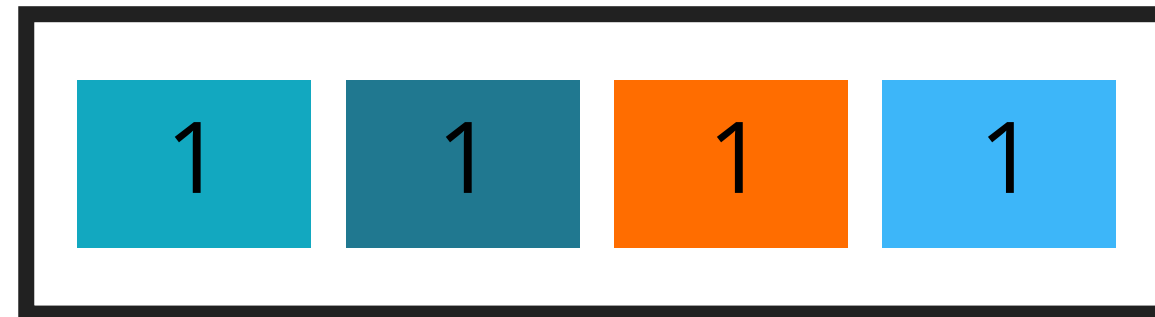
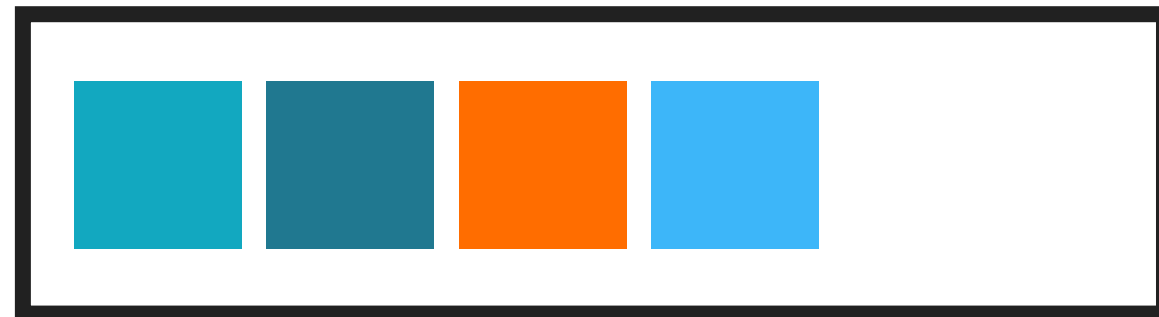


CSS layout

flex-grow

Met grow kun je aangeven hoeveel ruimte flex items krijgen in verhouding tot elkaar.

default is 0 en krijgen de items de ruimte die ze nodig hebben. Bij 1 groeien de items zodat de resterende ruimte wordt opgevuld. Indien alle items een 1 krijgen dan wordt de resterende ruimte gelijkwaardig verdeeld over alle items. Krijgt een item een hoger getal dan de anderen items? dan zal hij in verhouding meer groeien.



CSS layout

flex-shrink

De mogelijkheid voor een item om te verkleinen indien nodig.

De default value is 1 en dit betekent dat het item kan verkleinen.

0 betekent dat het item niet kan verkleinen. Dit kan erg handig zijn bij een item die de originele grootte moet behouden naast een item die veel ruimte in beslag neemt zoals een stuk tekst.



CSS layout

align-self

Met align-self heb je de mogelijkheid om de align-items property van de flex-container te overschrijven.



CSS layout

Flexbox documentatie

- | <https://css-tricks.com/snippets/css/a-guide-to-flexbox/>
- | https://www.w3schools.com/css/css3_flexbox.asp
- | https://developer.mozilla.org/en-US/docs/Learn/CSS/CSS_layout/Flexbox

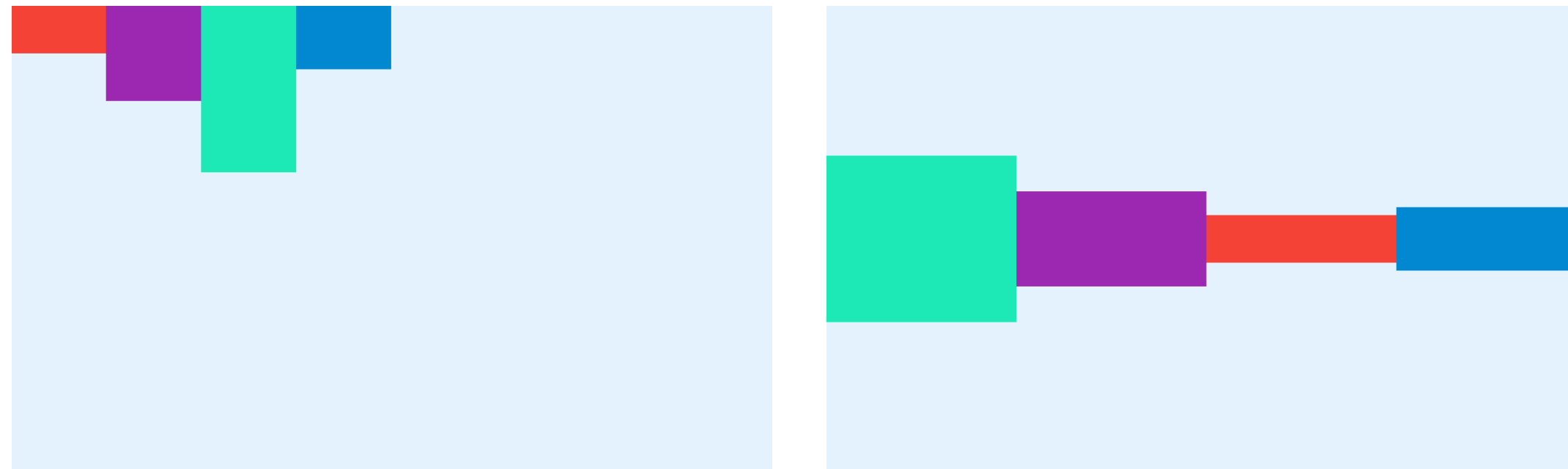
extra

property: flex-content
property: flex-basis
property: flex

Opdracht 1

Flexbox opdracht

Gebruik flexbox om de afbeeldingen na te maken. Download het HTML bestand en de afbeeldingen op github: <https://github.com/mauricedibbets1986/itvitae-/tree/main/les%203%20CSS%20intro/flexbox%20opdracht>



Opdracht 2

Persoonlijke pagina

Maak met de geleerde HTML en CSS technieken een persoonlijke pagina. Dit kan zijn over jezelf, een hobby of een idool.

Gebruik:

structuur -> header, main, footer, sections

headings en paragrafen

position absolute

flexbox